

Korrekte Software: Grundlagen und Methoden
Vorlesung 2 vom 10.04.24
Operationale Semantik

Serge Autexier, Christoph Lüth

Universität Bremen

Sommersemester 2024

Fahrplan

- ▶ Einführung
- ▶ Operationale Semantik
- ▶ Denotationale Semantik
- ▶ Äquivalenz der Operationalen und Denotationalen Semantik
- ▶ Der Floyd-Hoare-Kalkül
- ▶ Invarianten im Floyd-Hoare-Kalkül
- ▶ Korrektheit des Floyd-Hoare-Kalküls
- ▶ Strukturierte Datentypen
- ▶ Verifikationsbedingungen
- ▶ Vorwärts mit Floyd und Hoare
- ▶ Funktionen und Prozeduren I
- ▶ Funktionen und Prozeduren II
- ▶ Referenzen und Speichermodelle
- ▶ Ausblick und Rückblick

Zutaten

```
// GGT(A,B)
if (a == 0) r = b;
else {
  while (b != 0) {
    if (a <= b)
      b = b - a;
    else a = a - b;
  }
  r = a;
}
```

- ▶ Programme berechnen **Werte**
- ▶ Basierend auf
 - ▶ Werte sind **Variablen** zugewiesen
 - ▶ Evaluation von **Ausdrücken**
- ▶ Folgt dem Programmablauf

Unsere Programmiersprache

Wir betrachten einen Ausschnitt der Programmiersprache **C** (**C0**).

Ausbaustufe 1 kennt folgende Konstrukte:

- ▶ Typen: **int**;
- ▶ Ausdrücke: Variablen, Literale (für ganze Zahlen), arithmetische Operatoren (für ganze Zahlen), Relationen (**==**, **<**, ...), boolesche Operatoren (**&&**, **||**);
- ▶ Anweisungen:
 - ▶ Fallunterscheidung (**if**...**else**...), Iteration (**while**), Zuweisung, Blöcke;
 - ▶ Sequenzierung und leere Anweisung sind implizit

C0: Ausdrücke und Anweisungen

Aexp $a ::= \mathbf{Z} \mid \mathbf{Idt} \mid a_1 + a_2 \mid a_1 - a_2 \mid a_1 * a_2 \mid a_1 / a_2$

Bexp $b ::= \mathbf{1} \mid \mathbf{0} \mid a_1 == a_2 \mid a_1 < a_2 \mid !b \mid b_1 \&\& b_2 \mid b_1 \parallel b_2$

Exp $e ::= a \mid b$

Stmt $c ::= \mathbf{Idt} = \mathbf{Exp}$
| **if** (b) c_1 **else** c_2
| **while** (b) c
| $c_1; c_2$
| $\{ \}$

NB: Nicht die **konkrete** Syntax.

Was braucht die Semantik?

```
p = 1;  
c = 1;  
while (c <= n) {  
    p = p * c;  
    c = c + 1;  
}
```

- ▶ Ein Programm besteht aus **Anweisungen** und **Ausdrücken**.
- ▶ Ausdrücke werden **zustandsabhängig** ausgewertet.
- ▶ Anweisungen **überführen** Zustände.

Was braucht die Semantik?

```
p = 1;  
c = 1;  
while (c <= n) {  
    p = p * c;  
    c = c + 1;  
}
```

- ▶ Ein Programm besteht aus **Anweisungen** und **Ausdrücken**.
- ▶ Ausdrücke werden **zustandsabhängig** ausgewertet.
- ▶ Anweisungen **überführen** Zustände.

Woraus besteht die Semantik?

Was braucht die Semantik?

```
p = 1;  
c = 1;  
while (c <= n) {  
    p = p * c;  
    c = c + 1;  
}
```

- ▶ Ein Programm besteht aus **Anweisungen** und **Ausdrücken**.
- ▶ Ausdrücke werden **zustandsabhängig** ausgewertet.
- ▶ Anweisungen **überführen** Zustände.

Woraus besteht die Semantik?

- 1 Mathematische Modellierung des **Zustands**

Was braucht die Semantik?

```
p = 1;  
c = 1;  
while (c <= n) {  
    p = p * c;  
    c = c + 1;  
}
```

- ▶ Ein Programm besteht aus **Anweisungen** und **Ausdrücken**.
- ▶ Ausdrücke werden **zustandsabhängig** ausgewertet.
- ▶ Anweisungen **überführen** Zustände.

Woraus besteht die Semantik?

- ① Mathematische Modellierung des **Zustands**
- ② Auswertung von (arithmetischen und booleschen) Ausdrücken

Was braucht die Semantik?

```
p = 1;  
c = 1;  
while (c <= n) {  
    p = p * c;  
    c = c + 1;  
}
```

- ▶ Ein Programm besteht aus **Anweisungen** und **Ausdrücken**.
- ▶ Ausdrücke werden **zustandsabhängig** ausgewertet.
- ▶ Anweisungen **überführen** Zustände.

Woraus besteht die Semantik?

- ① Mathematische Modellierung des **Zustands**
- ② Auswertung von (arithmetischen und booleschen) Ausdrücken
- ③ Auswertung von Anweisungen: Zustandsübergänge

Semantik von C0

- ▶ Die (operationale) Semantik einer imperativen Sprache wie C0 ist ein **Zustandsübergang**: das System hat einen impliziten Zustand, der durch Zuweisung von **Werten** an **Adressen** geändert werden kann.

Systemzustände

- ▶ Ausdrücke werten zu **Werten** \mathbf{V} (hier ganze Zahlen) aus.
- ▶ Adressen **Loc** sind hier Programmvariablen (Namen): $\mathbf{Loc} = \mathbf{Idt}$
- ▶ Ein **Systemzustand** bildet Adressen auf Werte ab: $\Sigma = \mathbf{Loc} \rightarrow \mathbf{V}$
- ▶ Ein Programm bildet einen Anfangszustand **möglicherweise** auf einen Endzustand ab (wenn es **terminiert**).

Partielle, endliche Abbildungen I

Zustände sind **partielle, endliche Abbildungen** (finite partial maps)

$$f : X \rightarrow A$$

Notation:

- ▶ $f(x)$ für den Wert von x in f (*lookup*)
- ▶ $f(x) = \perp$ wenn x nicht in f (*undefined*)
- ▶ $f[x \mapsto n]$ für den Update an der Stelle x mit dem Wert n :

$$f[x \mapsto n](y) \stackrel{\text{def}}{=} \begin{cases} n & \text{if } x = y \\ f(y) & \text{otherwise} \end{cases}$$

Partielle, endliche Abbildungen II

Zustände sind **partielle, endliche Abbildungen** (finite partial maps)

$$f : X \rightarrow A$$

Notation:

- ▶ $\langle x \mapsto n, y \mapsto m \rangle$ u.ä. für konkrete Abbildungen.
- ▶ $\langle \rangle$ ist die leere (überall undefinierte Abbildung):

$$\text{für alle } x \in X \text{ gilt: } \langle \rangle(x) = \perp$$

- ▶ Die Domäne eines Zustands sind alle Stellen, an denen er definiert ist:

$$\text{Dom}(f) \stackrel{\text{def}}{=} \{x \in X \mid f(x) \neq \perp\}$$

- ▶ Updates sind “linksassoziativ”:

$$f[x \mapsto n][y \mapsto m] = (f[x \mapsto n])[y \mapsto m]$$

Arbeitsblatt 2.1: Zustände!

▶ Wie sieht ein Zustand aus, der a den Wert 6 und c den Wert 2 zuweist.

▶ Welches sind Zustände, und welche nicht:

A $\langle x \mapsto 1, a \mapsto 3 \rangle$

B $\langle x \mapsto y, b \mapsto 6 \rangle$

C $\langle x \mapsto 2, b \mapsto 6, x \mapsto 5 \rangle$

D $\langle x \mapsto 3, b \mapsto 6, y \mapsto 5 \rangle$

▶ Update von Zuständen:

A $\langle x \mapsto 1, a \mapsto 3 \rangle [y \mapsto 1] = ??$

B $\langle x \mapsto 1, a \mapsto 3 \rangle [x \mapsto 3] = ??$

C $\langle x \mapsto 1, a \mapsto 3 \rangle [x \mapsto 3][y \mapsto 1][x \mapsto 4] = ??$

Operationale Semantik: Arithmetische Ausdrücke

Ein arithmetischer Ausdruck a wertet unter Zustand σ zu einer ganzen Zahl n (Wert) aus.

$$\mathbf{Aexp} \ a ::= \mathbf{Z} \mid \mathbf{Idt} \mid a_1 + a_2 \mid a_1 - a_2 \mid a_1 * a_2 \mid a_1 / a_2 \quad \langle a, \sigma \rangle \rightarrow_{\mathbf{Aexp}} n$$

Operationale Semantik: Arithmetische Ausdrücke

Ein arithmetischer Ausdruck a wertet unter Zustand σ zu einer ganzen Zahl n (Wert) aus.

$$\mathbf{Aexp} \ a ::= \mathbf{Z} \mid \mathbf{Idt} \mid a_1 + a_2 \mid a_1 - a_2 \mid a_1 * a_2 \mid a_1 / a_2 \quad \langle a, \sigma \rangle \rightarrow_{Aexp} n$$

Regeln:

$$\frac{n \in \mathbf{Z}}{\langle n, \sigma \rangle \rightarrow_{Aexp} \llbracket n \rrbracket}$$

Operationale Semantik: Arithmetische Ausdrücke

Ein arithmetischer Ausdruck a wertet unter Zustand σ zu einer ganzen Zahl n (Wert) aus.

$$\mathbf{Aexp} \ a ::= \mathbf{Z} \mid \mathbf{Idt} \mid a_1 + a_2 \mid a_1 - a_2 \mid a_1 * a_2 \mid a_1 / a_2 \quad \langle a, \sigma \rangle \rightarrow_{Aexp} n$$

Regeln:

$$\frac{n \in \mathbf{Z}}{\langle n, \sigma \rangle \rightarrow_{Aexp} \llbracket n \rrbracket}$$

$$\frac{x \in \mathbf{Idt}, x \in \text{Dom}(\sigma), \sigma(x) = v}{\langle x, \sigma \rangle \rightarrow_{Aexp} v}$$

Operationale Semantik: Arithmetische Ausdrücke

Aexp $a ::= \mathbf{Z} \mid \mathbf{Idt} \mid a_1 + a_2 \mid a_1 - a_2 \mid a_1 * a_2 \mid a_1 / a_2 \quad \langle a, \sigma \rangle \rightarrow_{Aexp} n$

$$\frac{\langle a_1, \sigma \rangle \rightarrow_{Aexp} n_1 \quad \langle a_2, \sigma \rangle \rightarrow_{Aexp} n_2 \quad n_i \in \mathbb{Z}, n \text{ Summe } n_1 \text{ und } n_2}{\langle a_1 + a_2, \sigma \rangle \rightarrow_{Aexp} n}$$

$$\frac{\langle a_1, \sigma \rangle \rightarrow_{Aexp} n_1 \quad \langle a_2, \sigma \rangle \rightarrow_{Aexp} n_2 \quad n_i \in \mathbb{Z}, n \text{ Differenz } n_1 \text{ und } n_2}{\langle a_1 - a_2, \sigma \rangle \rightarrow_{Aexp} n}$$

$$\frac{\langle a_1, \sigma \rangle \rightarrow_{Aexp} n_1 \quad \langle a_2, \sigma \rangle \rightarrow_{Aexp} n_2 \quad n_i \in \mathbb{Z}, n \text{ Produkt } n_1 \text{ und } n_2}{\langle a_1 * a_2, \sigma \rangle \rightarrow_{Aexp} n}$$

$$\frac{\langle a_1, \sigma \rangle \rightarrow_{Aexp} n_1 \quad \langle a_2, \sigma \rangle \rightarrow_{Aexp} n_2 \quad n_i \in \mathbb{Z}, n_2 \neq 0, n \text{ Quotient } n_1, n_2}{\langle a_1 / a_2, \sigma \rangle \rightarrow_{Aexp} n}$$

Ableitungen

- ▶ Regeln werden von **unten** nach **oben** gelesen
- ▶ Regeln werden **komponiert** — es entsteht ein **Ableitungsbaum**

Beispiel: Auswertung von $x+3$ mit $\sigma \stackrel{def}{=} \langle x \mapsto 6, y \mapsto 5 \rangle$.

$$\frac{\langle x, \sigma \rangle \rightarrow_{Aexp} ? \quad \langle 3, \sigma \rangle \rightarrow_{Aexp} ?}{\langle x + 3, \sigma \rangle \rightarrow_{Aexp} ? + ?}$$

Ableitungen

- ▶ Regeln werden von **unten** nach **oben** gelesen
- ▶ Regeln werden **komponiert** — es entsteht ein **Ableitungsbaum**

Beispiel: Auswertung von $x+3$ mit $\sigma \stackrel{\text{def}}{=} \langle x \mapsto 6, y \mapsto 5 \rangle$.

$$\overline{\langle 3, \sigma \rangle \rightarrow_{Aexp} \llbracket 3 \rrbracket}$$

$$\frac{\langle x, \sigma \rangle \rightarrow_{Aexp} ? \quad \langle 3, \sigma \rangle \rightarrow_{Aexp} ?}{\langle x + 3, \sigma \rangle \rightarrow_{Aexp} ? + ?}$$

Ableitungen

- ▶ Regeln werden von **unten** nach **oben** gelesen
- ▶ Regeln werden **komponiert** — es entsteht ein **Ableitungsbaum**

Beispiel: Auswertung von $x+3$ mit $\sigma \stackrel{def}{=} \langle x \mapsto 6, y \mapsto 5 \rangle$.

$$\overline{\langle 3, \sigma \rangle \rightarrow_{Aexp} 3}$$

$$\frac{\langle x, \sigma \rangle \rightarrow_{Aexp} ? \quad \langle 3, \sigma \rangle \rightarrow_{Aexp} ?}{\langle x + 3, \sigma \rangle \rightarrow_{Aexp} ? + ?}$$

Ableitungen

- ▶ Regeln werden von **unten** nach **oben** gelesen
- ▶ Regeln werden **komponiert** — es entsteht ein **Ableitungsbaum**

Beispiel: Auswertung von $x+3$ mit $\sigma \stackrel{\text{def}}{=} \langle x \mapsto 6, y \mapsto 5 \rangle$.

$$\frac{x \in \text{dom}(\sigma), \sigma(x) = ?}{\langle x, \sigma \rangle \rightarrow_{Aexp} ?}$$

$$\frac{}{\langle 3, \sigma \rangle \rightarrow_{Aexp} 3}$$

$$\frac{\langle x, \sigma \rangle \rightarrow_{Aexp} ? \quad \langle 3, \sigma \rangle \rightarrow_{Aexp} ?}{\langle x + 3, \sigma \rangle \rightarrow_{Aexp} ? + ?}$$

Ableitungen

- ▶ Regeln werden von **unten** nach **oben** gelesen
- ▶ Regeln werden **komponiert** — es entsteht ein **Ableitungsbaum**

Beispiel: Auswertung von $x+3$ mit $\sigma \stackrel{\text{def}}{=} \langle x \mapsto 6, y \mapsto 5 \rangle$.

$$\frac{x \in \text{dom}(\sigma), \sigma(x) = 6}{\langle x, \sigma \rangle \rightarrow_{Aexp} 6}$$

$$\frac{}{\langle 3, \sigma \rangle \rightarrow_{Aexp} 3}$$

$$\frac{\langle x, \sigma \rangle \rightarrow_{Aexp} ? \quad \langle 3, \sigma \rangle \rightarrow_{Aexp} ?}{\langle x + 3, \sigma \rangle \rightarrow_{Aexp} ? + ?}$$

Ableitungen

- ▶ Regeln werden von **unten** nach **oben** gelesen
- ▶ Regeln werden **komponiert** — es entsteht ein **Ableitungsbaum**

Beispiel: Auswertung von $x+3$ mit $\sigma \stackrel{def}{=} \langle x \mapsto 6, y \mapsto 5 \rangle$.

$$\frac{}{\langle x, \sigma \rangle \rightarrow_{Aexp} 6} \qquad \frac{}{\langle 3, \sigma \rangle \rightarrow_{Aexp} 3}$$
$$\frac{\langle x, \sigma \rangle \rightarrow_{Aexp} ? \quad \langle 3, \sigma \rangle \rightarrow_{Aexp} ?}{\langle x + 3, \sigma \rangle \rightarrow_{Aexp} ? + ?}$$

Ableitungen

- ▶ Regeln werden von **unten** nach **oben** gelesen
- ▶ Regeln werden **komponiert** — es entsteht ein **Ableitungsbaum**

Beispiel: Auswertung von $x+3$ mit $\sigma \stackrel{\text{def}}{=} \langle x \mapsto 6, y \mapsto 5 \rangle$.

$$\frac{}{\langle x, \sigma \rangle \rightarrow_{Aexp} 6} \qquad \frac{}{\langle 3, \sigma \rangle \rightarrow_{Aexp} 3}$$
$$\frac{\langle x, \sigma \rangle \rightarrow_{Aexp} 6 \quad \langle 3, \sigma \rangle \rightarrow_{Aexp} 3}{\langle x + 3, \sigma \rangle \rightarrow_{Aexp} 6 + 3}$$

Ableitungen

- ▶ Regeln werden von **unten** nach **oben** gelesen
- ▶ Regeln werden **komponiert** — es entsteht ein **Ableitungsbaum**

Beispiel: Auswertung von $x+3$ mit $\sigma \stackrel{\text{def}}{=} \langle x \mapsto 6, y \mapsto 5 \rangle$.

$$\frac{}{\langle x, \sigma \rangle \rightarrow_{Aexp} 6} \qquad \frac{}{\langle 3, \sigma \rangle \rightarrow_{Aexp} 3}$$
$$\frac{\langle x, \sigma \rangle \rightarrow_{Aexp} 6 \quad \langle 3, \sigma \rangle \rightarrow_{Aexp} 3}{\langle x + 3, \sigma \rangle \rightarrow_{Aexp} 9}$$

Längere Beispiel-Ableitungen

Sei $\sigma \stackrel{\text{def}}{=} \langle x \mapsto 6, y \mapsto 5 \rangle$.

$$\overline{\langle (x + y) * (x - y), \sigma \rangle \rightarrow_{Aexp}}$$

Längere Beispiel-Ableitungen

Sei $\sigma \stackrel{\text{def}}{=} \langle x \mapsto 6, y \mapsto 5 \rangle$.

$$\frac{\overline{\langle x + y, \sigma \rangle \rightarrow_{Aexp}} \quad \overline{\langle x - y, \sigma \rangle \rightarrow_{Aexp}}}{\overline{\langle (x + y) * (x - y), \sigma \rangle \rightarrow_{Aexp}}}$$

Längere Beispiel-Ableitungen

Sei $\sigma \stackrel{\text{def}}{=} \langle x \mapsto 6, y \mapsto 5 \rangle$.

$$\frac{\frac{x \in \text{dom}(\sigma), \sigma(x) = 6}{\langle x, \sigma \rangle \rightarrow_{Aexp} 6}}{\langle x + y, \sigma \rangle \rightarrow_{Aexp}} \quad \frac{}{\langle x - y, \sigma \rangle \rightarrow_{Aexp}}$$

$$\langle (x + y) * (x - y), \sigma \rangle \rightarrow_{Aexp}$$

Längere Beispiel-Ableitungen

Sei $\sigma \stackrel{\text{def}}{=} \langle x \mapsto 6, y \mapsto 5 \rangle$.

$$\frac{\frac{x \in \text{dom}(\sigma), \sigma(x) = 6}{\langle x, \sigma \rangle \rightarrow_{Aexp} 6} \quad \frac{y \in \text{dom}(\sigma), \sigma(y) = 5}{\langle y, \sigma \rangle \rightarrow_{Aexp} 5}}{\langle x + y, \sigma \rangle \rightarrow_{Aexp}} \quad \frac{}{\langle x - y, \sigma \rangle \rightarrow_{Aexp}}$$

$$\langle (x + y) * (x - y), \sigma \rangle \rightarrow_{Aexp}$$

Längere Beispiel-Ableitungen

Sei $\sigma \stackrel{\text{def}}{=} \langle x \mapsto 6, y \mapsto 5 \rangle$.

$$\frac{\frac{x \in \text{dom}(\sigma), \sigma(x) = 6}{\langle x, \sigma \rangle \rightarrow_{Aexp} 6} \quad \frac{y \in \text{dom}(\sigma), \sigma(y) = 5}{\langle y, \sigma \rangle \rightarrow_{Aexp} 5}}{\langle x + y, \sigma \rangle \rightarrow_{Aexp} 11} \quad \frac{}{\langle x - y, \sigma \rangle \rightarrow_{Aexp} 5}}{\langle (x + y) * (x - y), \sigma \rangle \rightarrow_{Aexp} 11}$$

Längere Beispiel-Ableitungen

Sei $\sigma \stackrel{\text{def}}{=} \langle x \mapsto 6, y \mapsto 5 \rangle$.

$$\frac{\frac{x \in \text{dom}(\sigma), \sigma(x) = 6}{\langle x, \sigma \rangle \rightarrow_{Aexp} 6} \quad \frac{y \in \text{dom}(\sigma), \sigma(y) = 5}{\langle y, \sigma \rangle \rightarrow_{Aexp} 5}}{\langle x + y, \sigma \rangle \rightarrow_{Aexp} 11} \quad \frac{\frac{x \in \text{dom}(\sigma), \sigma(x) = 6}{\langle x, \sigma \rangle \rightarrow_{Aexp} 6} \quad \frac{y \in \text{dom}(\sigma), \sigma(y) = 5}{\langle y, \sigma \rangle \rightarrow_{Aexp} 5}}{\langle x - y, \sigma \rangle \rightarrow_{Aexp} 1}$$

$$\langle (x + y) * (x - y), \sigma \rangle \rightarrow_{Aexp} 11$$

Längere Beispiel-Ableitungen

Sei $\sigma \stackrel{\text{def}}{=} \langle x \mapsto 6, y \mapsto 5 \rangle$.

$$\frac{\frac{x \in \text{dom}(\sigma), \sigma(x) = 6}{\langle x, \sigma \rangle \rightarrow_{Aexp} 6} \quad \frac{y \in \text{dom}(\sigma), \sigma(y) = 5}{\langle y, \sigma \rangle \rightarrow_{Aexp} 5}}{\langle x + y, \sigma \rangle \rightarrow_{Aexp} 11} \quad \frac{\frac{x \in \text{dom}(\sigma), \sigma(x) = 6}{\langle x, \sigma \rangle \rightarrow_{Aexp} 6} \quad \frac{y \in \text{dom}(\sigma), \sigma(y) = 5}{\langle y, \sigma \rangle \rightarrow_{Aexp} 5}}{\langle x - y, \sigma \rangle \rightarrow_{Aexp} 1}$$

$$\langle (x + y) * (x - y), \sigma \rangle \rightarrow_{Aexp}$$

Längere Beispiel-Ableitungen

Sei $\sigma \stackrel{\text{def}}{=} \langle x \mapsto 6, y \mapsto 5 \rangle$.

$$\frac{\frac{x \in \text{dom}(\sigma), \sigma(x) = 6}{\langle x, \sigma \rangle \rightarrow_{Aexp} 6} \quad \frac{y \in \text{dom}(\sigma), \sigma(y) = 5}{\langle y, \sigma \rangle \rightarrow_{Aexp} 5}}{\langle x + y, \sigma \rangle \rightarrow_{Aexp} 11} \quad \frac{\frac{x \in \text{dom}(\sigma), \sigma(x) = 6}{\langle x, \sigma \rangle \rightarrow_{Aexp} 6} \quad \frac{y \in \text{dom}(\sigma), \sigma(y) = 5}{\langle y, \sigma \rangle \rightarrow_{Aexp} 5}}{\langle x - y, \sigma \rangle \rightarrow_{Aexp} 1}$$

$$\langle (x + y) * (x - y), \sigma \rangle \rightarrow_{Aexp} 11$$

Längere Beispiel-Ableitungen

Sei $\sigma \stackrel{\text{def}}{=} \langle x \mapsto 6, y \mapsto 5 \rangle$.

$$\frac{\frac{x \in \text{dom}(\sigma), \sigma(x) = 6}{\langle x, \sigma \rangle \rightarrow_{Aexp} 6} \quad \frac{y \in \text{dom}(\sigma), \sigma(y) = 5}{\langle y, \sigma \rangle \rightarrow_{Aexp} 5}}{\langle x + y, \sigma \rangle \rightarrow_{Aexp} 11} \quad \frac{\frac{x \in \text{dom}(\sigma), \sigma(x) = 6}{\langle x, \sigma \rangle \rightarrow_{Aexp} 6} \quad \frac{y \in \text{dom}(\sigma), \sigma(y) = 5}{\langle y, \sigma \rangle \rightarrow_{Aexp} 5}}{\langle x - y, \sigma \rangle \rightarrow_{Aexp} 1}$$

$$\langle (x + y) * (x - y), \sigma \rangle \rightarrow_{Aexp} 11$$

$$\frac{}{\langle (x * x) - (y * y), \sigma \rangle \rightarrow_{Aexp}}$$

Längere Beispiel-Ableitungen

Sei $\sigma \stackrel{\text{def}}{=} \langle x \mapsto 6, y \mapsto 5 \rangle$.

$$\frac{\frac{x \in \text{dom}(\sigma), \sigma(x) = 6}{\langle x, \sigma \rangle \rightarrow_{Aexp} 6} \quad \frac{y \in \text{dom}(\sigma), \sigma(y) = 5}{\langle y, \sigma \rangle \rightarrow_{Aexp} 5}}{\langle x + y, \sigma \rangle \rightarrow_{Aexp} 11} \quad \frac{\frac{x \in \text{dom}(\sigma), \sigma(x) = 6}{\langle x, \sigma \rangle \rightarrow_{Aexp} 6} \quad \frac{y \in \text{dom}(\sigma), \sigma(y) = 5}{\langle y, \sigma \rangle \rightarrow_{Aexp} 5}}{\langle x - y, \sigma \rangle \rightarrow_{Aexp} 1}$$

$$\langle (x + y) * (x - y), \sigma \rangle \rightarrow_{Aexp} 11$$

$$\frac{\frac{\langle x, \sigma \rangle \rightarrow_{Aexp} 6 \quad \langle x, \sigma \rangle \rightarrow_{Aexp} 6}{\langle x * x, \sigma \rangle \rightarrow_{Aexp} 36}}{\langle (x * x) - (y * y), \sigma \rangle \rightarrow_{Aexp}}$$

Längere Beispiel-Ableitungen

Sei $\sigma \stackrel{\text{def}}{=} \langle x \mapsto 6, y \mapsto 5 \rangle$.

$$\frac{\frac{x \in \text{dom}(\sigma), \sigma(x) = 6}{\langle x, \sigma \rangle \rightarrow_{Aexp} 6} \quad \frac{y \in \text{dom}(\sigma), \sigma(y) = 5}{\langle y, \sigma \rangle \rightarrow_{Aexp} 5}}{\langle x + y, \sigma \rangle \rightarrow_{Aexp} 11} \quad \frac{\frac{x \in \text{dom}(\sigma), \sigma(x) = 6}{\langle x, \sigma \rangle \rightarrow_{Aexp} 6} \quad \frac{y \in \text{dom}(\sigma), \sigma(y) = 5}{\langle y, \sigma \rangle \rightarrow_{Aexp} 5}}{\langle x - y, \sigma \rangle \rightarrow_{Aexp} 1}$$

$$\langle (x + y) * (x - y), \sigma \rangle \rightarrow_{Aexp} 11$$

$$\frac{\frac{\langle x, \sigma \rangle \rightarrow_{Aexp} 6 \quad \langle x, \sigma \rangle \rightarrow_{Aexp} 6}{\langle x * x, \sigma \rangle \rightarrow_{Aexp} 36} \quad \frac{\langle y, \sigma \rangle \rightarrow_{Aexp} 5 \quad \langle y, \sigma \rangle \rightarrow_{Aexp} 5}{\langle y * y, \sigma \rangle \rightarrow_{Aexp} 25}}{\langle (x * x) - (y * y), \sigma \rangle \rightarrow_{Aexp}}$$

Längere Beispiel-Ableitungen

Sei $\sigma \stackrel{\text{def}}{=} \langle x \mapsto 6, y \mapsto 5 \rangle$.

$$\frac{\frac{x \in \text{dom}(\sigma), \sigma(x) = 6}{\langle x, \sigma \rangle \rightarrow_{Aexp} 6} \quad \frac{y \in \text{dom}(\sigma), \sigma(y) = 5}{\langle y, \sigma \rangle \rightarrow_{Aexp} 5}}{\langle x + y, \sigma \rangle \rightarrow_{Aexp} 11} \quad \frac{\frac{x \in \text{dom}(\sigma), \sigma(x) = 6}{\langle x, \sigma \rangle \rightarrow_{Aexp} 6} \quad \frac{y \in \text{dom}(\sigma), \sigma(y) = 5}{\langle y, \sigma \rangle \rightarrow_{Aexp} 5}}{\langle x - y, \sigma \rangle \rightarrow_{Aexp} 1}$$

$$\langle (x + y) * (x - y), \sigma \rangle \rightarrow_{Aexp} 11$$

$$\frac{\frac{\langle x, \sigma \rangle \rightarrow_{Aexp} 6 \quad \langle x, \sigma \rangle \rightarrow_{Aexp} 6}{\langle x * x, \sigma \rangle \rightarrow_{Aexp} 36} \quad \frac{\langle y, \sigma \rangle \rightarrow_{Aexp} 5 \quad \langle y, \sigma \rangle \rightarrow_{Aexp} 5}{\langle y * y, \sigma \rangle \rightarrow_{Aexp} 25}}{\langle (x * x) - (y * y), \sigma \rangle \rightarrow_{Aexp} 11}$$

Arbeitsblatt 2.2: Auswertung

Konstruiert wie oben die Ableitung für den Ausdruck $(3*a)/b$ mit $\sigma \stackrel{\text{def}}{=} \langle a \mapsto 8, b \mapsto 7 \rangle$.

Hinweis: wahrscheinlich einfacher auf Papier...

Eigenschaften der Semantik

- ▶ **Frage:** Gegeben einen Ausdruck a , leitet **jeder** Zustand σ zu einem Wert n ab?

Eigenschaften der Semantik

- ▶ **Frage:** Gegeben einen Ausdruck a , leitet **jeder** Zustand σ zu einem Wert n ab?
- ▶ **Antwort:** Nein.
- ▶ Betrachte folgende Beispiele für $a \stackrel{\text{def}}{=} y+3/x$

$$\langle a, \langle y \mapsto 5 \rangle \rangle \rightarrow_{Aexp} ??? \quad (1)$$

$$\langle a, \langle y \mapsto 5, x \mapsto 0 \rangle \rangle \rightarrow_{Aexp} ??? \quad (2)$$

- ▶ In diesen Beispielen läßt sich kein **vollständiger** Ableitungsbaum konstruieren.
- ▶ Die Auswertung ist **undefiniert** — die Semantik ist **partiell**.

Operationale Semantik: Boolesche Ausdrücke

- **Bexp** $b ::= \mathbf{1} \mid \mathbf{0} \mid a_1 == a_2 \mid a_1 < a_2 \mid !b \mid b_1 \&\& b_2 \mid b_1 \parallel b_2$
 $\langle b, \sigma \rangle \rightarrow_{Bexp} true \mid false$

Regeln:

$$\frac{}{\langle \mathbf{1}, \sigma \rangle \rightarrow_{Bexp} true}$$

$$\frac{}{\langle \mathbf{0}, \sigma \rangle \rightarrow_{Bexp} false}$$

$$\frac{\langle a_1, \sigma \rangle \rightarrow_{Aexp} n_1 \quad \langle a_2, \sigma \rangle \rightarrow_{Aexp} n_2 \quad n_1 \text{ und } n_2 \text{ gleich}}{\langle a_1 == a_2, \sigma \rangle \rightarrow_{Bexp} true}$$

$$\frac{\langle a_1, \sigma \rangle \rightarrow_{Aexp} n_1 \quad \langle a_2, \sigma \rangle \rightarrow_{Aexp} n_2 \quad n_1 \text{ und } n_2 \text{ ungleich}}{\langle a_1 == a_2, \sigma \rangle \rightarrow_{Bexp} false}$$

Operationale Semantik: Boolesche Ausdrücke

- **Bexp** $b ::= \mathbf{1} \mid \mathbf{0} \mid a_1 == a_2 \mid a_1 < a_2 \mid !b \mid b_1 \&\& b_2 \mid b_1 \parallel b_2$
 $\langle b, \sigma \rangle \rightarrow_{Bexp} true \mid false$

Regeln:

$$\frac{\langle b, \sigma \rangle \rightarrow_{Bexp} true}{\langle !b, \sigma \rangle \rightarrow_{Bexp} false}$$

$$\frac{\langle b, \sigma \rangle \rightarrow_{Bexp} false}{\langle !b, \sigma \rangle \rightarrow_{Bexp} true}$$

$$\frac{\langle b_1, \sigma \rangle \rightarrow_{Bexp} false}{\langle b_1 \&\& b_2, \sigma \rangle \rightarrow_{Bexp} false}$$

$$\frac{\langle b_1, \sigma \rangle \rightarrow_{Bexp} true \quad \langle b_2, \sigma \rangle \rightarrow_{Bexp} t}{\langle b_1 \&\& b_2, \sigma \rangle \rightarrow_{Bexp} t}$$

$$\frac{\langle b_1, \sigma \rangle \rightarrow_{Bexp} true}{\langle b_1 \parallel b_2, \sigma \rangle \rightarrow_{Bexp} true}$$

$$\frac{\langle b_1, \sigma \rangle \rightarrow_{Bexp} false \quad \langle b_2, \sigma \rangle \rightarrow_{Bexp} t}{\langle b_1 \parallel b_2, \sigma \rangle \rightarrow_{Bexp} t}$$

Arbeitsblatt 2.3: Boolesche Ausdrücke

Konstruiert die Auswertung des Ausdrucks $b = x == 7 \ \&\& \ y == 3$ unter folgenden Zuständen:

① $\sigma_1 \stackrel{def}{=} \langle x \mapsto 7, y \mapsto 3 \rangle$

② $\sigma_2 \stackrel{def}{=} \langle x \mapsto 6, y \mapsto 3 \rangle$

③ $\sigma_3 \stackrel{def}{=} \langle y \mapsto 6 \rangle$

④ $\sigma_4 \stackrel{def}{=} \langle x \mapsto 7 \rangle$

⑤ $\sigma_5 \stackrel{def}{=} \langle x \mapsto 2 \rangle$

Striktheit

- ▶ Eine partielle Funktion f ist **strikt** wenn $f(x)$ undefiniert ist, sobald x undefiniert ist.
- ▶ In unserer Semantik sind alle Operatoren (arithmetisch und boolesch) strikt, **bis auf** `&&` und `||` im **ersten** Argument.
 - ▶ Operational nennt man das auch abgekürzte Auswertung (*short-circuit evaluation*)
 - ▶ Das erlaubt Idiome wie `if (x != 0 && 3/x > 1) { ... }`
- ▶ Wie erkennt man Striktheit an den **Regeln**?

Striktheit

- ▶ Eine partielle Funktion f ist **strikt** wenn $f(x)$ undefiniert ist, sobald x undefiniert ist.
- ▶ In unserer Semantik sind alle Operatoren (arithmetisch und boolesch) strikt, **bis auf** `&&` und `||` im **ersten** Argument.
 - ▶ Operational nennt man das auch abgekürzte Auswertung (*short-circuit evaluation*)
 - ▶ Das erlaubt Idiome wie `if (x != 0 && 3/x > 1) { ... }`
- ▶ Wie erkennt man Striktheit an den **Regeln**?
Alle Variablen der Konklusion kommen in den Bedingungen vor.

Operationale Semantik: Anweisungen

► **Stmt** $c ::= \text{Idt} = \text{Exp} \mid \text{if } (b) \ c_1 \ \text{else } \ c_2 \mid \text{while } (b) \ c \mid c_1; c_2 \mid \{ \}$

Beispiel:

$$\langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'$$

$$\langle x = 5, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'$$

wobei $\sigma'(x) = 5$ und $\sigma'(y) = \sigma(y)$ für alle $y \neq x$

Operationale Semantik: Anweisungen

► **Stmt** $c ::= \text{Idt} = \text{Exp} \mid \text{if } (b) \ c_1 \ \text{else } \ c_2 \mid \text{while } (b) \ c \mid c_1; c_2 \mid \{ \}$

Beispiel:

$$\langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'$$

$$\langle x = 5, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'$$

wobei $\sigma'(x) = 5$ und $\sigma'(y) = \sigma(y)$ für alle $y \neq x$
bzw. $\sigma' \stackrel{\text{def}}{=} \sigma[x \mapsto 5]$

Operationale Semantik: Anweisungen

► Stmt $c ::= \text{Idt} = \text{Exp} \mid \text{if } (b) \ c_1 \ \text{else} \ c_2 \mid \text{while } (b) \ c \mid c_1; c_2 \mid \{ \}$

Beispiel:

$$\langle c, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'$$

$$\langle x = 5, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma[x \mapsto 5]$$

wobei $\sigma'(x) = 5$ und $\sigma'(y) = \sigma(y)$ für alle $y \neq x$
bzw. $\sigma' \stackrel{\text{def}}{=} \sigma[x \mapsto 5]$

Operationale Semantik: Anweisungen

► **Stmt** $c ::= \text{Idt} = \text{Exp} \mid \text{if } (b) \ c_1 \ \text{else} \ c_2 \mid \text{while } (b) \ c \mid c_1; c_2 \mid \{ \}$

Regeln:

$$\frac{}{\langle \{ \}, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma} \qquad \frac{\langle a, \sigma \rangle \rightarrow_{\text{Aexp}} n \in \mathbb{Z}}{\langle x = a, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma[x \mapsto n]}$$

$$\frac{\langle c_1, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma' \quad \langle c_2, \sigma' \rangle \rightarrow_{\text{Stmt}} \sigma''}{\langle c_1; c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma''}$$

$$\frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{true} \quad \langle c_1, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}{\langle \text{if } (b) \ c_1 \ \text{else} \ c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'} \qquad \frac{\langle b, \sigma \rangle \rightarrow_{\text{Bexp}} \text{false} \quad \langle c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}{\langle \text{if } (b) \ c_1 \ \text{else} \ c_2, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'}$$

Operationale Semantik: Anweisungen

► Stmt $c ::= \text{Idt} = \text{Exp} \mid \text{if } (b) \ c_1 \ \text{else} \ c_2 \mid \text{while } (b) \ c \mid c_1; c_2 \mid \{ \}$

Regeln:

$$\frac{\langle b, \sigma \rangle \rightarrow_{Bexp} \text{false}}{\langle \text{while } (b) \ c, \sigma \rangle \rightarrow_{Stmt} \sigma}$$

$$\frac{\langle b, \sigma \rangle \rightarrow_{Bexp} \text{true} \quad \langle c, \sigma \rangle \rightarrow_{Stmt} \sigma' \quad \langle \text{while } (b) \ c, \sigma' \rangle \rightarrow_{Stmt} \sigma''}{\langle \text{while } (b) \ c, \sigma \rangle \rightarrow_{Stmt} \sigma''}$$

Beispiel

```
x = 1;  
while (y != 0) {  
  y = y - 1;  
  x = 2 * x;  
}  
// x = 2y
```

$\sigma \stackrel{\text{def}}{=} \langle y \mapsto 2 \rangle$

$$\frac{\frac{\langle 1, \sigma \rangle \rightarrow_{Aexp} 1}{\langle x = 1, \sigma \rangle \rightarrow_{Stmt} \sigma[x \mapsto 1] := \sigma_1} \quad \frac{\frac{\langle y, \sigma_1 \rangle \rightarrow_{Aexp} 2}{\langle y! = 0, \sigma_1 \rangle \rightarrow_{Bexp} true} \quad \frac{(A)}{\langle y = y - 1; x = 2 * x, \sigma_1 \rangle \rightarrow_{Stmt?}} \quad \frac{(B)}{\langle w, ? \rangle \rightarrow_{Stmt?}}}{\langle \mathbf{while} (y! = 0) \{y = y - 1; x = 2 * x\}, \sigma_1 \rangle \rightarrow_{Stmt?}}}{\langle x = 1; \underbrace{\mathbf{while} (y! = 0) \{y = y - 1; x = 2 * x\}}_w, \sigma \rangle \rightarrow_{Stmt?}}$$

(A)

$$\frac{\frac{\langle y - 1, \sigma_1 \rangle \rightarrow_{Aexp} 1}{\langle y = y - 1, \sigma_1 \rangle \rightarrow_{Stmt} \sigma_1[y \mapsto 1] := \sigma_2} \quad \frac{\langle 2 * x, \sigma_2 \rangle \rightarrow_{Aexp} 2}{\langle x = 2 * x, \sigma_2 \rangle \rightarrow_{Stmt} \sigma_2[x \mapsto 2] := \sigma_3}}{\langle y = y - 1; x = 2 * x, \sigma_1 \rangle \rightarrow_{Stmt} \sigma_3}$$

$$\frac{\langle 1, \sigma \rangle \rightarrow_{Aexp} 1}{\langle x = 1, \sigma \rangle \rightarrow_{Stmt} \sigma_1} \quad \frac{\frac{\langle y, \sigma_1 \rangle \rightarrow_{Aexp} 2}{\langle y! = 0, \sigma_1 \rangle \rightarrow_{Bexp} true} \quad \frac{\text{(A)}}{\langle y = y - 1; x = 2 * x, \sigma_1 \rangle \rightarrow_{Stmt} \sigma_3} \quad \frac{\text{(B)}}{\langle w, \sigma_3 \rangle \rightarrow_{Stmt} ?}}{\langle \mathbf{while} (y! = 0) \{y = y - 1; x = 2 * x\}, \sigma_1 \rangle \rightarrow_{Stmt} ?}}$$

$$\langle x = 1; \underbrace{\mathbf{while} (y! = 0) \{y = y - 1; x = 2 * x\}}_w, \sigma \rangle \rightarrow_{Stmt} ?$$

(B)

$$\frac{\frac{\langle y, \sigma_3 \rangle \rightarrow_{Aexp} 1}{\langle y! = 0, \sigma_3 \rangle \rightarrow_{Bexp} true} \quad \frac{\frac{\langle y - 1, \sigma_3 \rangle \rightarrow_{Aexp} 0}{\langle y = y - 1, \sigma_3 \rangle \rightarrow_{Stmt} \sigma_3[y \mapsto 0]} \quad \frac{\langle 2 * x, \sigma_4 \rangle \rightarrow_{Aexp} 4}{\langle x = 2 * x, \sigma_4 \rangle \rightarrow_{Stmt} \sigma_4[x \mapsto 4]} := \sigma_5}{\langle y = y - 1; x = 2 * x, \sigma_3 \rangle \rightarrow_{Stmt} \sigma_5} \quad (C)}{\langle w, \sigma_3 \rangle \rightarrow_{Stmt} \sigma_5} \quad (C)$$

$$\left. \begin{array}{l} \frac{\langle y, \sigma_5 \rangle \rightarrow_{Aexp} 0}{\langle y! = 0, \sigma_3 \rangle \rightarrow_{Bexp} false} \\ \frac{\langle w, \sigma_5 \rangle \rightarrow_{Stmt} \sigma_5}{} \end{array} \right\} (C)$$

while $(y! = 0)$ $\{y = y - 1; x = 2 * x\}$
 w

$$\begin{array}{c}
 \frac{\langle y, \sigma_1 \rangle \rightarrow_{Aexp} 2}{\langle y! = 0, \sigma_1 \rangle \rightarrow_{Bexp} true} \quad \frac{(A)}{\langle y = y - 1; x = 2 * x, \sigma_1 \rangle \rightarrow_{Stmt} \sigma_3} \quad \frac{(B)}{\langle w, \sigma_3 \rangle \rightarrow_{Stmt} \sigma_5} \\
 \dots \quad \frac{\langle \mathbf{while} (y! = 0) \{y = y - 1; x = 2 * x\}, \sigma_1 \rangle \rightarrow_{Stmt} \sigma_5}{\langle x = 1; \underbrace{\mathbf{while} (y! = 0) \{y = y - 1; x = 2 * x\}}_w, \sigma \rangle \rightarrow_{Stmt} \sigma_5}
 \end{array}$$

$$\begin{aligned}
 \sigma_5 &= \sigma_4[x \mapsto 4] = \sigma_3[y \mapsto 0][x \mapsto 4] = \sigma_2[x \mapsto 2][y \mapsto 0][x \mapsto 4] \\
 &= \sigma_1[y \mapsto 1][x \mapsto 2][y \mapsto 0][x \mapsto 4] = \langle y \mapsto 2 \rangle [y \mapsto 1][x \mapsto 2][y \mapsto 0][x \mapsto 4] \\
 &= \langle y \mapsto 0, x \mapsto 4 \rangle
 \end{aligned}$$

und es gilt $\sigma_5(x) = 4 = 2^2 = 2^{\sigma_1(y)}$

Lineare, abgekürzte Schreibweise

```
// ⟨y ↦ 2⟩  
x = 1;  
//  
while (y != 0) {  
  y = y - 1;  
  x = 2 * x;  
}
```

Lineare, abgekürzte Schreibweise

```
// ⟨y ↦ 2⟩  
x = 1;  
// ⟨y ↦ 2, x ↦ 1⟩  
while (y != 0) {  
  y = y - 1;  
  x = 2 * x;  
}
```

Lineare, abgekürzte Schreibweise

```
// ⟨y ↦ 2⟩
x = 1; // Ableitung für ⟨x = 1, ⟨y ↦ 2⟩⟩ →Stmt ⟨y ↦ 2, x ↦ 1⟩
// ⟨y ↦ 2, x ↦ 1⟩
while (y != 0) // ⟨y != 0, ⟨y ↦ 2, x ↦ 1⟩⟩ →Bexp true
|           y = y - 1; // Ableitung für ⟨y = y - 1, ⟨y ↦ 2, x ↦ 1⟩⟩ →Stmt ⟨y ↦
1, x ↦ 1⟩
|           // ⟨y ↦ 1, x ↦ 1⟩
|           x = 2 * x; // Ableitung für ⟨x = 2 * x, ⟨y ↦ 1, x ↦ 1⟩⟩ →Stmt ...
|           // ⟨y ↦ 1, x ↦ 2⟩
while (y != 0) {
  y = y - 1;
  x = 2 * x;
}
```

Lineare, abgekürzte Schreibweise

```
//  $\langle y \mapsto 2 \rangle$   
x = 1;  
//  $\langle y \mapsto 2, x \mapsto 1 \rangle$   
while (y!=0) //  $\langle y! = 0, \langle y \mapsto 2, x \mapsto 1 \rangle \rangle \rightarrow_{Bexp} true$   
|     y = y - 1; // Ableitung für  $y = y - 1$   
|     //  $\langle y \mapsto 1, x \mapsto 1 \rangle$   
|     x = 2 * x; // Ableitung für  $x = 2 * x$   
|     //  $\langle y \mapsto 1, x \mapsto 2 \rangle$   
while (y!=0) //  $\langle y! = 0, \langle y \mapsto 1, x \mapsto 2 \rangle \rangle \rightarrow_{Bexp} true$   
|     y = y - 1;  
|     //  $\langle y \mapsto 0, x \mapsto 2 \rangle$   
|     x = 2 * x;  
|     //  $\langle y \mapsto 0, x \mapsto 4 \rangle$   
while (y!=0) //  $\langle y! = 0, \langle y \mapsto 0, x \mapsto 4 \rangle \rangle \rightarrow_{Bexp} false$   
//  $\langle y \mapsto 0, x \mapsto 4 \rangle$ 
```

Was haben wir gezeigt?

```
//  $\langle y \mapsto 2 \rangle$   $\sigma_1$   
x = 1;  
//  $\langle y \mapsto 2, x \mapsto 1 \rangle$   
while (y != 0) {  
  y = y - 1;  
  x = 2 * x;  
}  
//  $\langle y \mapsto 0, x \mapsto 4 \rangle$   $\sigma_E$ 
```

- Für einen festen Anfangszustand $\sigma_1 = \langle y \mapsto 2 \rangle$ gilt am Ende $\sigma_E(x) = 4 = 2^2 = 2^{\sigma_1(y)}$.

Was haben wir gezeigt?

```
// ⟨y ↦ 2⟩                σ1  
x = 1;  
// ⟨y ↦ 2, x ↦ 1⟩  
while (y != 0) {  
  y = y - 1;  
  x = 2 * x;  
}  
// ⟨y ↦ 0, x ↦ 4⟩        σE
```

- ▶ Für **einen festen Anfangszustand** $\sigma_1 = \langle y \mapsto 2 \rangle$ gilt am Ende $\sigma_E(x) = 4 = 2^2 = 2^{\sigma_1(y)}$.
- ▶ Gilt das für alle?

Was haben wir gezeigt?

```
// ⟨y ↦ 2⟩                                 $\sigma_1$   
x = 1;  
// ⟨y ↦ 2, x ↦ 1⟩  
while (y != 0) {  
  y = y - 1;  
  x = 2 * x;  
}  
// ⟨y ↦ 0, x ↦ 4⟩                             $\sigma_E$ 
```

- ▶ Für **einen festen Anfangszustand** $\sigma_1 = \langle y \mapsto 2 \rangle$ gilt am Ende $\sigma_E(x) = 4 = 2^2 = 2^{\sigma_1(y)}$.
- ▶ Gilt das für alle?
- ▶ Für welche nicht?

Was haben wir gezeigt?

```
// ⟨y ↦ 2⟩                                 $\sigma_1$   
x = 1;  
// ⟨y ↦ 2, x ↦ 1⟩  
while (y != 0) {  
  y = y - 1;  
  x = 2 * x;  
}  
// ⟨y ↦ 0, x ↦ 4⟩                           $\sigma_E$ 
```

- ▶ Für **einen festen Anfangszustand** $\sigma_1 = \langle y \mapsto 2 \rangle$ gilt am Ende $\sigma_E(x) = 4 = 2^2 = 2^{\sigma_1(y)}$.
- ▶ Gilt das für alle?
- ▶ Für welche nicht?
- ▶ Wie kann man das für alle Anfangs-Zustände, für die es gilt, zeigen?

Was passiert hier?

```
//  $\langle y \mapsto -1 \rangle$   
x = 1;  
while (y != 0) {  
  y = y - 1;  
  x = 2 * x;  
}
```

Was passiert hier?

```
// ⟨y ↦ -1⟩  
x = 1;  
while (y != 0) {  
  y = y - 1;  
  x = 2 * x;  
}
```

- ▶ Ableitung terminiert nicht (Ableitungsbaum der Auswertung der while-Schleife wächst unendlich)
- ▶ In linearer Schreibweise geht es immer wieder unten weiter.

Arbeitsblatt 2.4: Programme!

- ▶ Werten Sie das nebenstehende Programm aus für den Anfangszustand $\langle x \mapsto 5, y \mapsto 2 \rangle$
- ▶ Geben Sie die Auswertung in abgekürzter Schreibweise an.
- ▶ Welche Beziehung gilt am Ende des Programms zwischen den Werten von x und y im Endzustand und im Anfangszustand?

```
while (y != 0) {  
  x = x * x;  
  y = y - 1;  
}
```

Äquivalenz arithmetischer Ausdrücke

Gegeben zwei Aexp a_1 and a_2

- Sind sie gleich?

$$a_1 \sim_{Aexp} a_2 \text{ gdw } \forall \sigma, n. \langle a_1, \sigma \rangle \rightarrow_{Aexp} n \Leftrightarrow \langle a_2, \sigma \rangle \rightarrow_{Aexp} n$$

$$(x*x) + 2*x*y + (y*y) \quad \text{und} \quad (x+y) * (x+y)$$

- Wann sind sie gleich?

$$\forall \sigma, n. \langle a_1, \sigma \rangle \rightarrow_{Aexp} n \Leftrightarrow \langle a_2, \sigma \rangle \rightarrow_{Aexp} n$$

$$\begin{array}{l} x*x \quad \text{und} \quad 8*x+9 \\ x*x \quad \text{und} \quad x*x+1 \end{array}$$

Äquivalenz Boolescher Ausdrücke

Gegeben zwei Bexp-Ausdrücke b_1 and b_2

► Sind sie gleich?

$$b_1 \sim_{Bexp} b_2 \text{ iff } \forall \sigma, b. \langle b_1, \sigma \rangle \rightarrow_{Bexp} b \Leftrightarrow \langle b_2, \sigma \rangle \rightarrow_{Bexp} b$$

A || (A && B) und A

Beweisen

Zwei Programme c_0, c_1 sind äquivalent gdw. sie die gleichen Zustandsveränderungen bewirken. Formal definieren wir

Definition

$$c_0 \sim c_1 \text{ iff } \forall \sigma, \sigma'. \langle c_0, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma' \Leftrightarrow \langle c_1, \sigma \rangle \rightarrow_{\text{Stmt}} \sigma'$$

Ein einfaches Beispiel:

Lemma

Sei $w \equiv \mathbf{while} (b) c$ mit $b \in \mathbf{Bexp}$, $c \in \mathbf{Stmt}$.

Dann gilt: $w \sim \mathbf{if} (b) \{c; w\} \mathbf{else} \{\}$

Beweis

- ▶ Gegeben beliebiger Programmzustand σ .
- ▶ **Zu zeigen:** sowohl w also auch **if** (b) $\{c; w\}$ **else** $\{\}$ werten zum gleichen Programmzustand aus (wenn sie auswerten).
- ▶ Der Beweis geht per Fallunterscheidung über die Auswertung von Teilausdrücken bzw. Teilprogrammen.

Beweis

① $\langle b, \sigma \rangle \rightarrow_{Bexp} \text{false}$:

$$\begin{aligned} & \langle \text{while } (b) \ c, \sigma \rangle \rightarrow_{Stmt} \sigma \\ \langle \text{if } (b) \ \{c; w\} \ \text{else } \{\}, \sigma \rangle & \rightarrow_{Stmt} \langle \{\}, \sigma \rangle \rightarrow_{Stmt} \sigma \end{aligned}$$

Beweis

① $\langle b, \sigma \rangle \rightarrow_{Bexp} \text{false}$:

$$\begin{aligned} & \langle \text{while } (b) \ c, \sigma \rangle \rightarrow_{Stmt} \sigma \\ \langle \text{if } (b) \ \{c; w\} \ \text{else } \{\}, \sigma \rangle & \rightarrow_{Stmt} \langle \{\}, \sigma \rangle \rightarrow_{Stmt} \sigma \end{aligned}$$

② $\langle b, \sigma \rangle \rightarrow_{Bexp} \text{true}$: Sei $\langle c, \sigma \rangle \rightarrow_{Stmt} \sigma'$, dann:

$$\begin{aligned} & \overbrace{\langle \text{while } (b) \ c, \sigma \rangle}^w \rightarrow_{Stmt} \langle c, \sigma \rangle \rightarrow_{Stmt} \sigma' \\ & \langle w, \sigma' \rangle \rightarrow_{Stmt} \sigma'' \\ \langle \text{if } (b) \ \{c; w\} \ \text{else } \{\}, \sigma \rangle & \rightarrow_{Stmt} \langle \{c; w\}, \sigma \rangle \rightarrow_{Stmt} \langle c, \sigma \rangle \rightarrow_{Stmt} \sigma' \\ & \langle w, \sigma' \rangle \rightarrow_{Stmt} \sigma'' \end{aligned}$$

Beweis

① $\langle b, \sigma \rangle \rightarrow_{Bexp} \text{false}$:

$$\begin{aligned} & \langle \text{while } (b) \ c, \sigma \rangle \rightarrow_{Stmt} \sigma \\ \langle \text{if } (b) \ \{c; w\} \ \text{else } \{\}, \sigma \rangle & \rightarrow_{Stmt} \langle \{\}, \sigma \rangle \rightarrow_{Stmt} \sigma \end{aligned}$$

② $\langle b, \sigma \rangle \rightarrow_{Bexp} \text{true}$: Sei $\langle c, \sigma \rangle \rightarrow_{Stmt} \sigma'$, dann:

$$\begin{aligned} & \overbrace{\langle \text{while } (b) \ c, \sigma \rangle}^w \rightarrow_{Stmt} \langle c, \sigma \rangle \rightarrow_{Stmt} \sigma' \\ & \langle w, \sigma' \rangle \rightarrow_{Stmt} \sigma'' \\ \langle \text{if } (b) \ \{c; w\} \ \text{else } \{\}, \sigma \rangle & \rightarrow_{Stmt} \langle \{c; w\}, \sigma \rangle \rightarrow_{Stmt} \langle c, \sigma \rangle \rightarrow_{Stmt} \sigma' \\ & \langle w, \sigma' \rangle \rightarrow_{Stmt} \sigma'' \end{aligned}$$

③ $\langle b, \sigma \rangle$ wertet gar nicht aus — dann werten weder w noch $\text{if } (b) \ \{c; w\} \ \text{else } \{\}$ aus.

Zusammenfassung

- ▶ Operationale Semantik als ein Mittel zur Beschreibung der Semantik
- ▶ Auswertungsregeln:
 - ▶ arbeiten entlang der syntaktischen Struktur
 - ▶ werten (zu gegebenem Zustand) Ausdrücke zu Werten aus (Zahlen, Booleschen Werten)
 - ▶ und (zu gegebenem Zustand) Programme zu Zuständen
- ▶ Fragen zu Programmen: Gleichheit